REMARKS

The Examiner has objected to the disclosure for the use of 'byte'. More specifically, the Examiner indicates that "The use of byte in the specification and claims is ambiguous and indefinite since paragraph [0073] refers to bytes comprising 9-bits contrary to the standard meaning of the term." (Office Action of 4/10/08, Page 3.)

It is well established that the patentee can be his own lexicographer. (*ZMI Corp. v. Cardiac Resuscitator Corp.*, 844 F.2d 1576, 6 USPQ2d 1557 (Fed. Cir. 1988).) In describing the number of bits in a byte, the Applicant states in paragraph [0079]:

> In this scheme, 8 bits in each byte can be used to carry data. The other bit in each byte can be grouped together to carry the EDC code. As illustrated in Figure 9, for an 8-byte data packet, each byte can be used to carry 8 bits of data and 1 bit of the 8 bit EDC code. … Those skilled in the art may recognize that the number of bits in a byte, the number of EDC bits in a byte and the number of bytes in a data packet can be chosen rather arbitrarily. For instance, a four byte packet with each byte containing 18 bits can be used. Then two bits in each byte can be used to carry a portion of the EDC code. (Emphasis added.)

Thus, the number of bits in a byte is not critical to the Applicants invention. Moreover, the Applicants do not understand how describing an example wherein a byte has 9-bits could possibly render the term 'byte' ambiguous or indefinite. In view of the description provided in paragraph [0079], the Applicants believe that the use of the term 'byte' in the specification is neither ambiguous nor indefinite. For these reasons, the Applicants request that the Examiner's objection be dismissed.

Claims 1-4 have been rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to

particularly point out and distinctly claim the subject matter which the Applicant regards as the invention.

More specifically, the Examiner states it is not clear what is meant by "defining each byte of the packet to have an EDC code portion" as recited by Claim 1. (Office Action of 4/10/08, Page 3.)

In order to understand the above-cited portion of Claim 1, the entire corresponding element of Claim 1 should be considered. This claim element recites "defining each byte of the packet to have an EDC code portion and a data portion, wherein each EDC code portion is a distributed portion of a complete EDC code".

In accordance with one embodiment of the Applicants invention, this recited element of Claim 1 may be represented by Figure 9 of the Application as originally filed (which is reproduced below). Figure 9 illustrates an 8-byte packet, wherein each byte of the packet is used to carry 8 bits of data and 1 bit of an 8-bit EDC code. The EDC code associated with the packet is represented by the bits ED_0-ED_7, which are distributed among the 8 bytes of the packet. (Specification, paragraph [0079].)

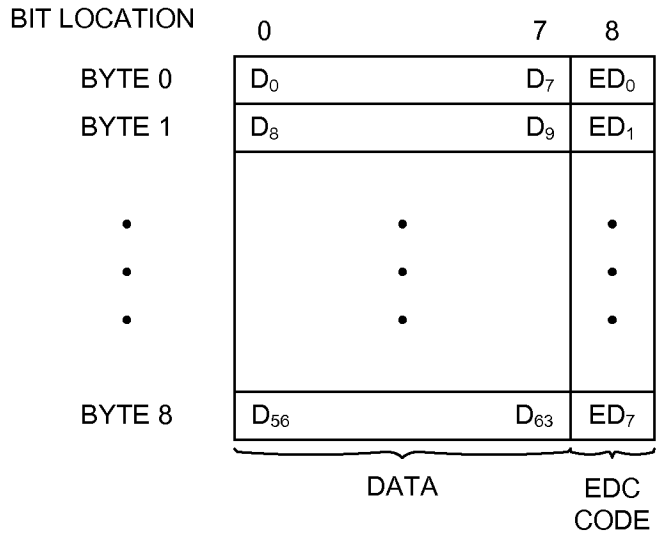

FIG. 9

Thus, in accordance with one embodiment of the present invention, including a bit of the EDC code in each of the 8 bytes of the packet corresponds with 'defining each byte of the packet to have an EDC code portion' as recited by Claim 1. For this reason, the Applicants believe that Claim 1 meets the requirements of 35 U.S.C. 112, second paragraph. The Applicants therefore request that the rejection of Claims 1-4 under 35 U.S.C. 112, second paragraph, be withdrawn.

In rejecting the claims under 35 U.S.C. 112, second paragraph, the Examiner further states:

> a packet P either has an EDC portion or it doesn't. the Examiner *assumes* the following: -- generating and EDC code portion for each byte of a packet so that each byte has a data portion and an EDC portion, wherein each EDC code portion is a distributed portion of a complete EDC code". (Emphasis added.) (Office Action of 4/10/08, Page 3.)

Although the Applicants are not sure what relevance

this statement has (if any), the Applicants note that the Examiner's assumption is faulty for the following reasons. A packet may have a complete EDC code, wherein portions of the complete EDC code are not distributed among the various bytes of the packet. Rather, the complete EDC code may be provided as an extra byte, in addition to the various bytes of the packet. In fact, this configuration is suggested by Ragle, which teaches that a 9-bit error check character E is computed for a seven character word, wherein the error check character E forms an eighth character of the word. (Ragle, Col. 3, lines 59-68; Fig. 1.)

In rejecting the claims under 35 U.S.C. 112, second paragraph, the Examiner further indicates that "The use of byte in the specification and the claims is ambiguous and indefinite since paragraph [0073] refers to bytes comprising 9-bits contrary to the standard meaning of the term". (Office Action of 4/10/08, Page 4.) However, as described above, the term 'byte' is neither ambiguous nor indefinite. Thus, the use of the term 'byte' in the claims does not prevent Claims 1-4 from meeting the requirements of 35 U.S.C. 112, second paragraph.

Claims 1 and 4 have been rejected under 35 U.S.C. 102(b) as being anticipated by Ragle (U.S. Patent 4,052,698) in view of Sako (U.S. Patent 4,788,685).

Claim 1 recites "defining each byte of the packet to have an EDC code portion and a data portion, wherein each EDC code portion is a distributed portion of a complete EDC code", "storing said data portion and said EDC code portion of each byte of the packet in the memory module" and "reading out said data portion and said EDC code portion of each byte of the packet from said memory module".

Paragraph [0080] of the Application as originally filed

specifies an advantage of these steps as follows, "The destined memory module stores both the EDC code and data indiscriminately, in other words it simply stores the whole packet in the cache or in the memory core without further data processing." (Emphasis added.) "When no error is detected as is true most of the time, EDC operations has little effect on the memory accessing time".

Ragle teaches that a parity bit P is computed for each of seven 8-bit characters (i.e., bytes) D1-D7, and that an error check character E is computed for the seven character word, wherein the error check character forms an eighth character of the word. (Ragle, Col. 3, lines 59-68.) However, Ragle teaches that the resulting 8-bit by 9-bit matrix must be converted to a 10-bit by 9-bit matrix before recording onto tape 102. (Ragle, Col. 4, lines 1-4.) The conversion required by Ragle undesirably requires a separate encoder 110. (Ragle, Col. 4, lines 5-23; Figs. 1-3.) The conversion required by Ragle also undesirably requires a larger memory, because the conversion increases the number of required storage bits. The conversion required by Ragle also undesirably increases the memory access time.

However, Ragle teaches that this conversion is required to provide "no more than two adjacent zeros", and to "never [have] more than one zero leading or ending a code". (Ragle, Col. 4, lines 15-23.) Fig. 2 of Ragle illustrates the conversion of an 8-bit by 9-bit matrix (which includes parity bits P and error check character E) to a 10-bit by 9-bit matrix (which includes generic bits X).

By teaching that this conversion is necessary, Ragle explicitly teaches away from storing the parity bits P and the error check character E on the tape 102. Ragle therefore fails to teach "storing said data portion and said

EDC code portion of each byte of the packet in the memory module" as recited by amended Claim 1. Because Ragle fails to teach "storing said data portion and said EDC code portion of each byte of the packet in the memory module" as recited by amended Claim 1, Ragle also necessarily fails to teach "reading out said data portion and said EDC code portion of each byte of the packet from said memory module" as recited by amended Claim 1.

In rejecting the Applicants arguments, the Examiner contends that "Figure 1 in Ragle teaches storing said data portion B1-B8 and said EDC code portion P of each 9-bit byte B1-B8, P of the word/packet in the magnetic tape memory module 102". (Office Action of 4/10/08, pages 4-5.) This is simply not correct. "The entire matrix 108 including parity and check bits" is an 8x9 array, which is labeled as a "DATA GROUP" in Figure 1 of Ragle. However, this 8x9 DATA GROUP is not written to the tape 102. Instead, Ragle requires that the 8x9 DATA GROUP must be converted into a 9x10 RECORD GROUP that is written to the tape 102. Ragle does not (and cannot) specify the locations of the data, parity and check bits in the converted 9x10 RECORD GROUP. Note that Figure 2 of Ragle shows the converted 9x10 RECORD GROUP with generic bits labeled 'X', while the nature of every bit in the 8x9 DATA GROUP of Fig. 2 is specified as data bit (B), parity bit (P) or check bit (B encircled with dashed lines).

The distinction between bits of the 8x9 DATA GROUP and bits of the 9x10 RECORD GROUP is illustrated by Figure 3 of Ragle. For example, a 4-bit value of '0000' present in the 8x9 DATA GROUP is converted to a 5-bit value '11001' in the 9x10 RECORD GROUP. Suppose that the least significant bit (0000) of the 4-bit value represents a check bit and the

three most significant bits (0000) of the 4-bit value represent data bits. Which bits of the converted 5-bit value '11001' represent the original check bit? Which bits of the converted 5-bit value '11001' represent the original data bits? While the converted 5-bit value '11001' may be representative of the data bits and check bit present in the original 4-bit value, it is clear that the original data bits and check bit are not written to tape 102. By teaching that the original 8x9 DATA GROUP must be converted to the 9x10 RECORD GROUP before being written to the tape 102, Ragle teach away from writing the original 8x9 DATA GROUP to the tape 102. Because the original 8x9 DATA GROUP is not written to the tape 102, the original 8x9 DATA GROUP necessarily cannot be read from the tape 102.

The Examiner agrees that "Ragle does not explicitly teach the specific use of bypassing modulation encoder 110 … to store the 8-character by 9-bit matrix". (Office Action, Page 6, lines 10-11.) However, the Examiner indicates that it would be obvious to bypass the modulation encoder 110 of Ragle in view of the teachings of Sako. More specifically, the Examiner indicates that Sako "teaches that the product code of Figure 5 is a sector format for data stored into a sector of memory". (Office Action, page 6, lines 13-14.) The Examiner further indicates that "it would have been obvious … to modify Ragle with the teachings of Sako by including use of bypassing modulation encoder 110 in Ragle to store the 8-character by 9bit matrix comprising the 8 9-bit bytes of B1-B8,P." (Office Action, page 6, lines 15-18.) The Applicants disagree for the following reasons.

First, Sako explicitly teaches that an ECC encoder 38 should be used, wherein "additional information and redundant data C1 and C2" is added to the data D0, D1, D2,

…. (Sako, Col. 5, lines 60-66.) Because Sako explicitly teaches the use of an ECC encoder 38, Sako teaches away from bypassing the modulation encoder 110 of Ragle as suggested by the Examiner.

In addition, the Examiner states that "This modification would have been obvious … because … bypassing modulation encoder 110 in Ragle … would have provided a storage means compliant with sector formatted data". (Emphasis added.) However, there is no teaching in either Sako or Ragle that would suggest that the output of the parity & check character generator 108 of Ragle would be "compliant with sector formatted data", while the output of modulation encoder 110 of Ragle would not be "compliant with sector formatted data". That is, there is no teaching that it is necessary to bypass the modulation encoder 110 of Ragle in order to provide "a storage means compliant with sector formatted data" as suggested by the Examiner. It would therefore not be obvious to modify Ragle to bypass the modulation encoder 110 for the purpose of obtaining "a storage means compliant with sector formatted data" as suggested by the Examiner.

For these reasons, Claim 1 is allowable over Ragle in view of Sako under 35 U.S.C. 103(a). Claim 4, which depends from Claim 1, is allowable over Ragle in view of Sako for at least the same reasons as Claim 1.

The Applicants therefore respectfully request reconsideration and withdrawal of the pending rejections of Claims 1 and 4 under 35 U.S.C. 103(a).

Claim 2 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Ragle in view of Sako and further in view of Brune (U.S. Patent 3,665,393). Claim 2, which depends from Claim 1, is allowable over Ragle in view of

Sako for at least the same reasons as Claim 1. Because Brune does not seem to remedy the above-described deficiencies of Ragle and Sako, Claim 2 is allowable over the combination of Ragle, Sako and Brune for at least the same reasons that Claim 1 is allowable over Ragle and Sako. The Applicants therefore respectfully request reconsideration and withdrawal of the pending rejection of Claim 2 under 35 U.S.C. 103.

Claim 3 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Ragle in view of Sako. As set forth above, independent Claim 1 is allowable over Ragle in view of Sako. Claim 3, which depends from Claim 1, is allowable over Ragle in view of Sako for at least the same reasons as Claim 1. The Applicants therefore respectfully request reconsideration and withdrawal of the pending rejection of Claim 3 under 35 U.S.C. 103.

The Applicants have added new Claim 12, which recites "A method as in claim 1, wherein said forwarding of said data portion is performed in parallel with said performing error checking and correction in said EDC functional block." (Emphasis added.) Support for this amendment is found in the application as originally filed at paragraph [0080]. No new matter is added. Note that Ragle teaches that "as the data group passes through correction unit 306 which may be a buffer register, it is corrected, if appropriate". (Ragle, Col. 11, Lines 10-12, 16-21, Fig. 7.) Thus, Ragle teaches that the data group is not forwarded until after errors are corrected.

The Applicants have also added new Claims 13 and 14, which are supported by paragraph [0080] and Fig. 10A of the Application as originally filed. No new matter is added.

CONCLUSION

Claims 1-4 and 12-14 are pending in the present Application. Reconsideration and allowance of these claims is respectfully requested. If the Examiner has any questions or comments, he is invited to call the undersigned.

Respectfully submitted,

/E. ERIC HOFFMAN/

Customer No. 86658

E. Eric Hoffman
Reg. No. 38,186
Attorney for Applicants
BEVER, HOFFMAN & HARMS, LLP

Tel. No.: (925) 895-3545
Fax No.: (408) 451-5908